

УДК 05.13.18

Препроцессор программной системы SIGMA для обучения разработчиков САЕ-приложений, используемых в САПР аэрокосмической отрасли.

Столярчук В.А.

*Московский авиационный институт (национальный исследовательский университет), МАИ, Волоколамское шоссе, 4, Москва, А-80, ГСП-3, 125993, Россия
e-mail:vladimir.stolyarchuk@gmail.com*

Аннотация

Представлено краткое описание препроцессора учебной программной системы, предназначенной для подготовки разработчиков систем инженерного анализа, используемых в САПР авиационно-космической техники. Назначение препроцессора – подготовка исходных данных для расчета. Объект расчета – плоская пластина. Препроцессор позволяет сформировать оптимальную сетку конечных элементов, используя разные методы триангуляции и оценить её качество до, собственно, самих вычислений.

Геометрическая модель рассчитываемого объекта создается с помощью стандартных примитивов типа: точка, прямая, дуга, окружность и разделяется на 8-ми узловые зоны. Формирование сетки конечных элементов проводится методом изопараметрических координат, фронтальным методом и несколькими методами Делоне. Для фронтального метода предусмотрено использование функции плотности, которая задается заранее в численном виде в интерактивном режиме с использованием графических элементов. В методах Делоне сначала генерируется

набор опорных узлов сетки конечных элементов (5 способов), затем проводится триангуляция области с использованием (на выбор) алгоритма S-Hull и метода Пола Бурка. Примеры работы всех алгоритмов триангуляции приведены. Дополнительно в препроцессоре имеется подсистема оптимизации сетки методом увеличения минимального угла и подсистема регуляризации сетки на основе позиционирования каждого узла по центру тяжести многоугольника. Имеется подсистема анализа качества сетки, позволяющий дать обзор полной статистики по сетке с графическим выделением областей, обладающих высоким и низким качеством сетки. В результате, препроцессор представляет собой развитый инструмент для формирования разных видов сеток и исследования их характеристик. Интерфейс программы разработан с учетом обучающих функций для пользователя. Графический интерфейс в системе реализован с использованием стандартной библиотеки WinForms. В разработанной системе используется технология MDI (Multi Document Interface), что позволяет одновременно работать с несколькими проектами. Сама разработка реализована на языке C# с использованием платформы .NET (разработка велась в IDE Microsoft Visual Studio 2010 Ultimate) и библиотека Tao Framework , предоставляющая разработчикам .NET и Mono доступ к возможностям популярных библиотек типа OpenGL и SDL.

В процессе эксплуатации препроцессора накоплен некоторый научный результат и получен большой методический опыт.

Ключевые слова: САЕ-система, метод конечных элементов, методы триангуляции, оптимизация сетки

Одним из недостатков любых коммерческих пакетов, в том числе, ANSYS, NASTRAN, PATRAN, SAMCEF, COSMOS и др. является закрытый код, что делает недоступным модификацию даже отдельных модулей. Это является серьезным препятствием при подготовке специалистов по разработке таких программных пакетов. Поэтому для обучения специалистов по разработке САЕ-систем (систем инженерного анализа) приходится создавать учебные системы МКЭ (метода конечных элементов), отличительная особенность которых состоит в том, что учебные системы, сохраняя основные функциональные свойства промышленных комплексов, имеют упрощенные алгоритмы системного и прикладного программного обеспечения и являются системами с открытым кодом. Понятно, что такие системы должны снабжаться достаточно развитой и весьма специфичной информационной базой, необходимой для процесса обучения, учитывая, что студентам приходится осваивать одновременно целый комплекс дисциплин.

В МАИ на кафедре «Прикладной информатики» разработана учебная система SIGMA [1], реализующая МКЭ для плоского напряженно-деформированного состояния и призванная решать задачу обучения принципам построения САЕ-систем [2]. Важной особенностью SIGMA является то, что при создании её компонентов активное участие принимают студенты 4-5-х курсов, дипломники и аспиранты кафедры. Работы студентов являются темами выступлений на конференциях, в том числе международных. SIGMA является для студентов, по сути, проектным программным объектом, на котором они познают особенности

разработки САЕ-систем и накапливают опыт, позволяющий им быстрее адаптироваться к запросам предприятий и учреждений аэрокосмической отрасли.

Задачи МКЭ крайне чувствительны к качеству сеток, на которых проводится решение. Именно поэтому как для обучения, так и в исследовательских целях для SIGMA разработан препроцессор, предоставляющий возможность генерировать сетки разными методами с последующей оценкой их качества до проведения собственно расчета. В создании препроцессора принимали участие несколько студентов кафедры, но основную работу по программированию выполнил студент Панфилов А.А. [3], которому кафедра и руководитель работ выражают искреннюю благодарность. Основными задачами, решаемыми препроцессором, являются:

- создание геометрической модели объекта, формирование сетки КЭ, задание граничных условий, приложение внешних воздействий, задание свойств КЭ;
- демонстрация работы различных методов построения сетки КЭ;
- анализ, оценка и оптимизация сетки КЭ;
- экспорт/импорт в/из SIGMA.

В препроцессоре реализованы методы триангуляции плоской области методом изопараметрических координат, фронтальным методом и триангуляцией Делоне (два алгоритма).

Дополнительно имеется возможность формирования и экспорта данных в подсистему триангуляции методами Рапперта [4]. Последние реализованы непосредственно в основной программе SIGMA.

В качестве обязательного элемента для всех методов триангуляции принято предварительное разбиение области на 4-х-узловые зоны.

В разработке препроцессора использовались язык C# с использованием платформы .NET (разработка велась в IDE Microsoft Visual Studio 2010 Ultimate) и библиотека Tao Framework.

Ниже (рис. 1) приведена схема взаимодействия основных модулей системы.

Остановимся на показательных элементах данной схемы.

Ядро препроцессора - это совокупность структур и методов, необходимых для обеспечения корректного взаимодействия всех модулей системы. Основные функции, возлагаемые на ядро:

- обмен данными между модулями и модулем хранения данных;
- хранение данных модели;



Рис. 1. Схема взаимодействия основных модулей системы

- организация взаимодействия между графическим интерфейсом пользователя и модулем хранения данных;
- организация ввода/вывода данных рассчитываемого объекта с использованием его графического отображения.
- обеспечение необходимых функций для работы с проектом: создание, удаление, сохранение, отмена/повтор внесенных изменений и т.п.

Основным классом, используемым в ядре, является класс FullModel.

Графический интерфейс в системе реализован с использованием стандартной библиотеки WinForms. В разработанной системе используется технология MDI (Multi Document Interface), что позволяет одновременно работать с несколькими

проектами. Ниже (рис. 2) приведен основной вид панели инструментов системы и пример (рис. 3) внешнего вида системы при работе с несколькими проектами.



Рис. 2. Внешний вид панели инструментов препроцессора

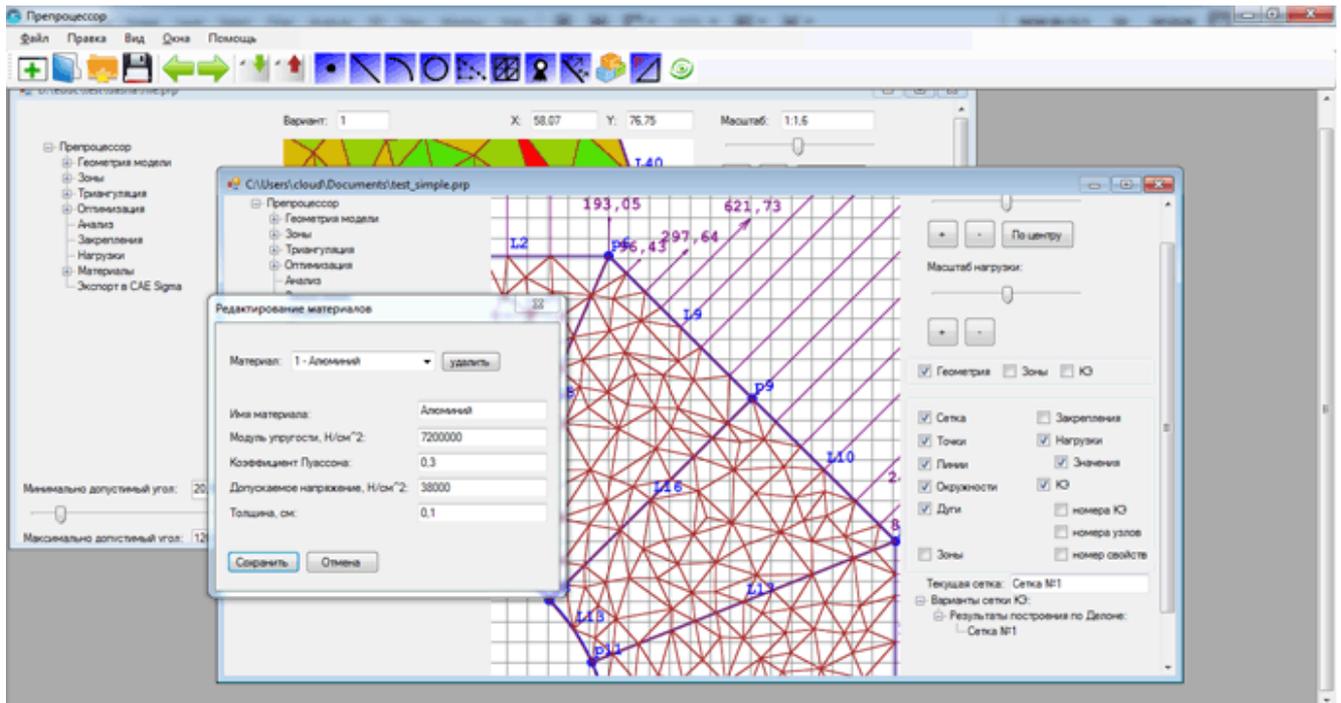


Рис.3. Организация рабочего пространства препроцессора при открытии нескольких окон.

Модуль построения геометрической модели пластины использует обычные геометрические примитивы такие, как точка, прямая, дуга, окружность и инструменты их модификации. Модуль формирования 8-ми узловых зон предоставляет инструменты для разбиения объекта на базовые зоны типов (рис.4):

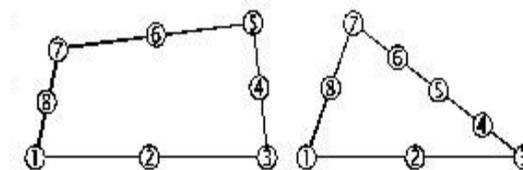


Рис.4.

Система обеспечивает формирование зон двумя методами. В первом случае пользователь указывает линии, образующие зону. Для этого используется управляющий элемент, либо выбираются линии кликом мыши по рабочему пространству (рис. 5). Во втором случае пользователь кликом мыши выбирает пространство внутри предполагаемой зоны, и алгоритм сам производит поиск линий.

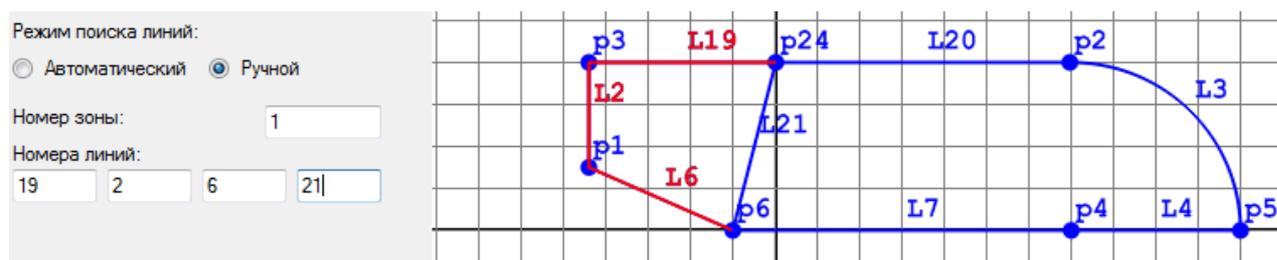


Рис.5. Формирование зон при ручном поиске линий

И в том и в другом случае система сообщает об ошибках формирования зоны. Результатом успешной работы того или иного метода станет создание зоны (рис.6):

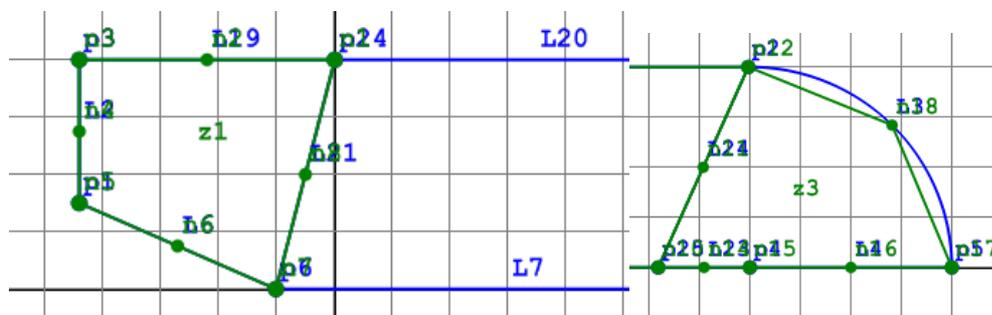


Рис. 6. Пример зоны в препроцессоре. Справа – аппроксимация дуги двумя отрезками.

Система автоматически добавляет промежуточные узлы на линии зон. В случае, когда одной из сторон зоны оказывается дуга, промежуточный угол автоматически образуется на дуге окружности. При помощи такого промежуточного узла, дуга аппроксимируется двумя отрезками:

Пример отображения зон в препроцессоре представлен слева (рис. 7). Справа (рис.8) – результат работы сеточного генератора методом изопараметрических координат:

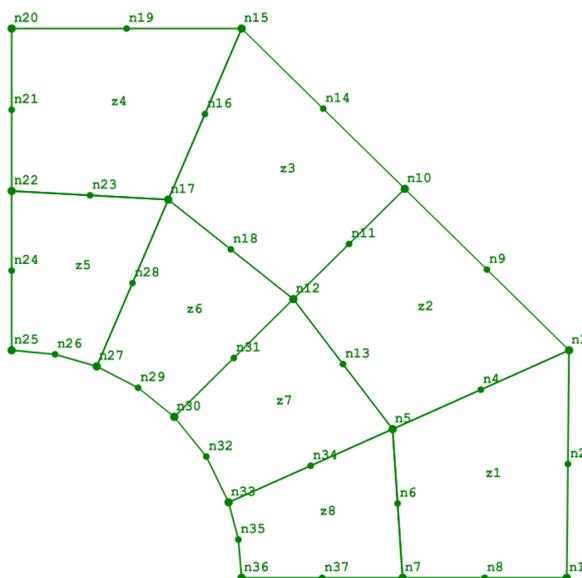


Рис.7. Отображение зон

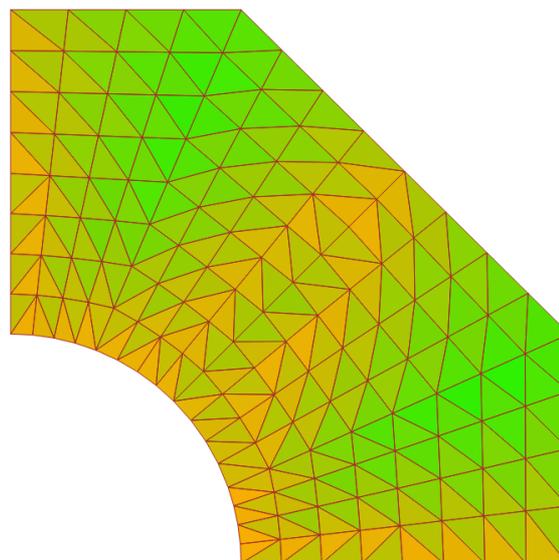


Рис.8. Результат построения сетки методом изопараметрических координат

Наиболее сложные модули триангуляции реализованы для фронтального метода и методов Делоне [5]. Фронтальный метод может работать с использованием и без использования функции плотности. На рис.9 представлен результат триангуляции фронтальным методом с использованием критерия «по двум сторонам с делением угла пополам». На границы зон были добавлены узлы с шагом 1.5 для получения более густой сетки КЭ. На рис. 10 представлена триангуляция фронтальным методом с использованием функции плотности. Функция плотности формируется заранее.

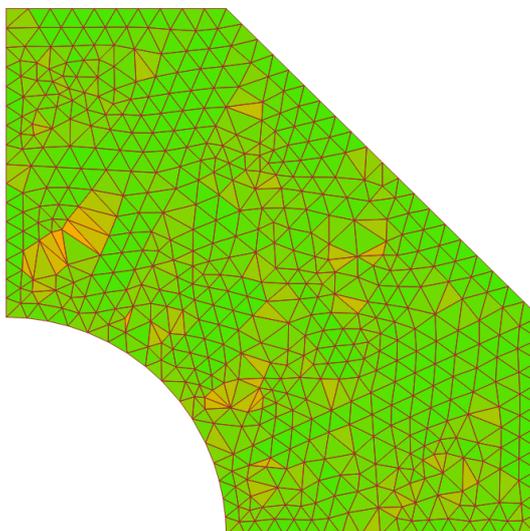


Рис.9. Результат построения сетки фронтальным методом без использования функции плотности

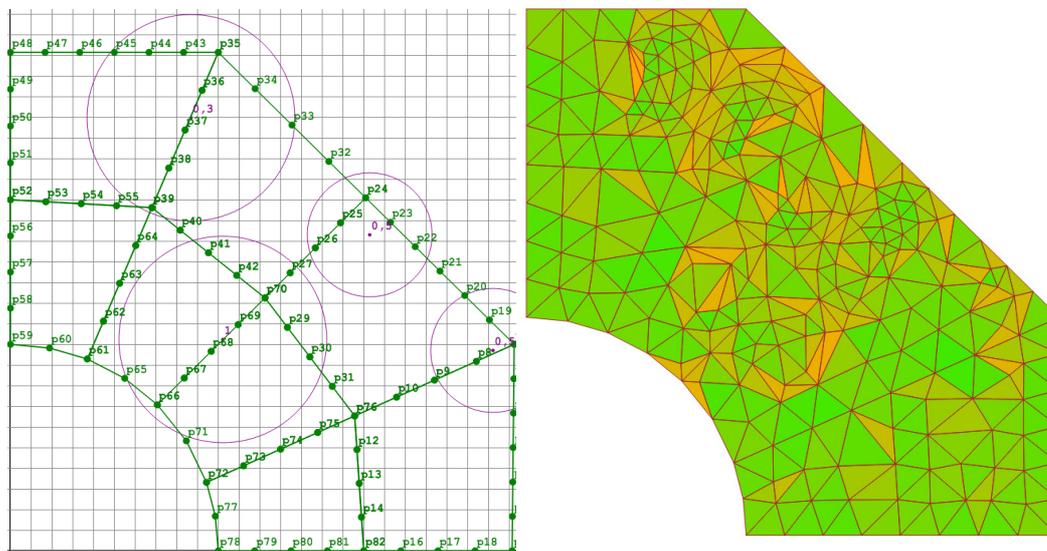


Рис.10. Результат построения сетки фронтальным методом с использованием функции плотности. Слева окружностями указаны области изменения функции плотности.

Триангуляции по Делоне в препроцессоре строятся в два этапа. Сначала генерируются опорные точки, затем подключается собственно алгоритм триангуляции. Решение первой задачи система предлагает одним из следующих методов: хаотическая генерация точек, хаотическая генерация точек с распределением плотности, прямоугольная регулярная структура точек, равноугольная регулярная структура точек.

Каждый из этих подходов направлен на построение множества точек в некоторой области.

Из методов триангуляции используются два: алгоритм S-Hull [6] и алгоритм Пола Бурка [7]. Первый опубликован в 2011 году, второй – в 1989-ом.

В качестве примеров на рис. 11-14 приведены триангуляции алгоритмом S-Hull.

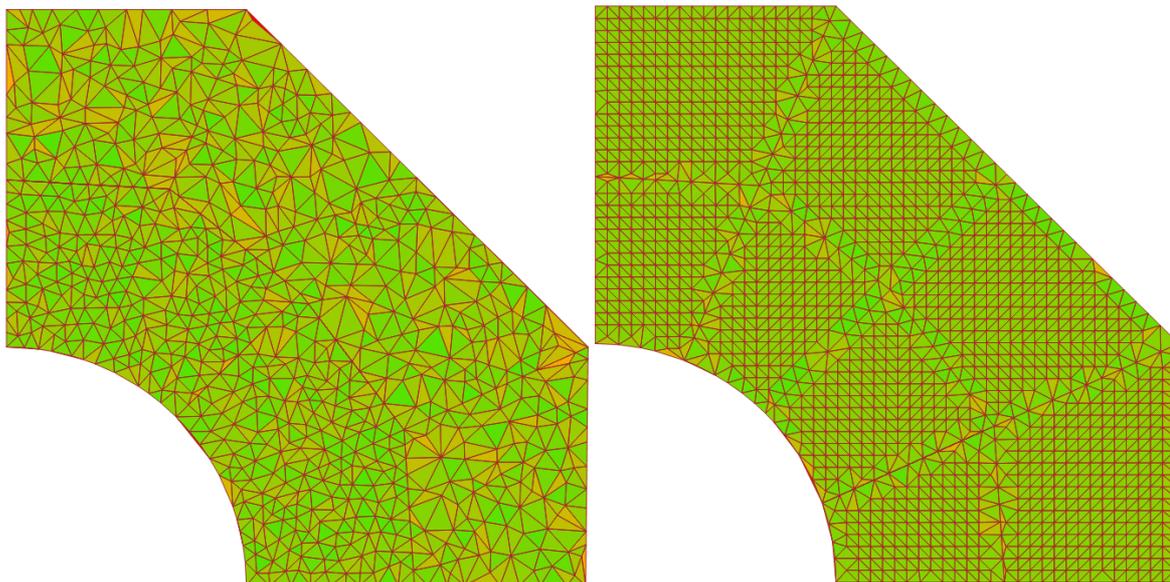


Рис. 11. Хаотическая генерация точек.

Рис. 12. Прямоугольная регулярная структура.

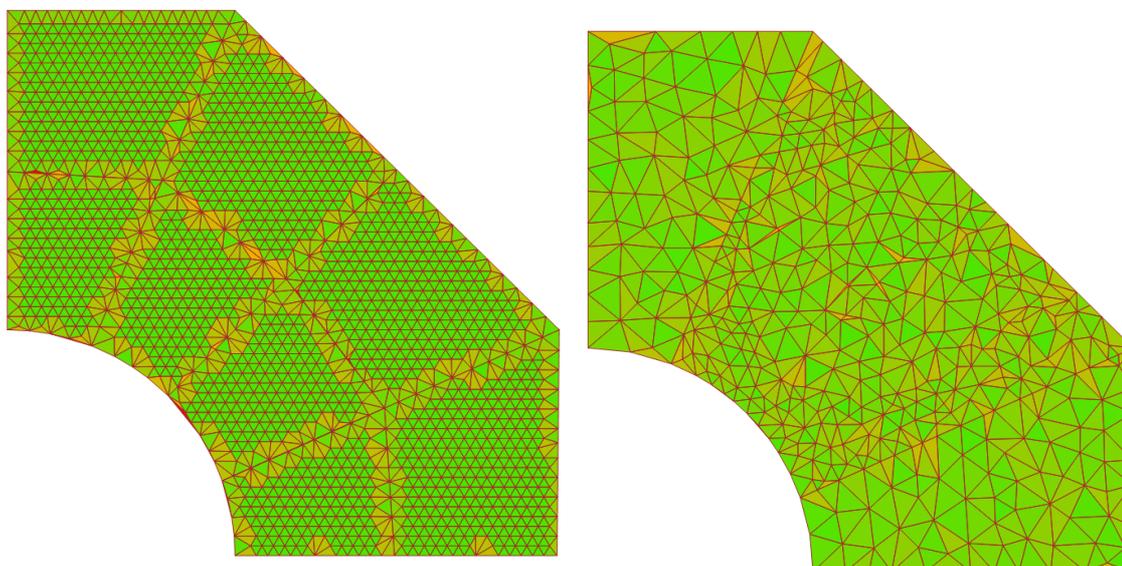


Рис.13. Равноугольная регулярная структура точек

Рис.14. Хаотическая генерация с распределением плотности.

Ниже (рис.15) приведена форма настроек метода и пример выделения подобласти:

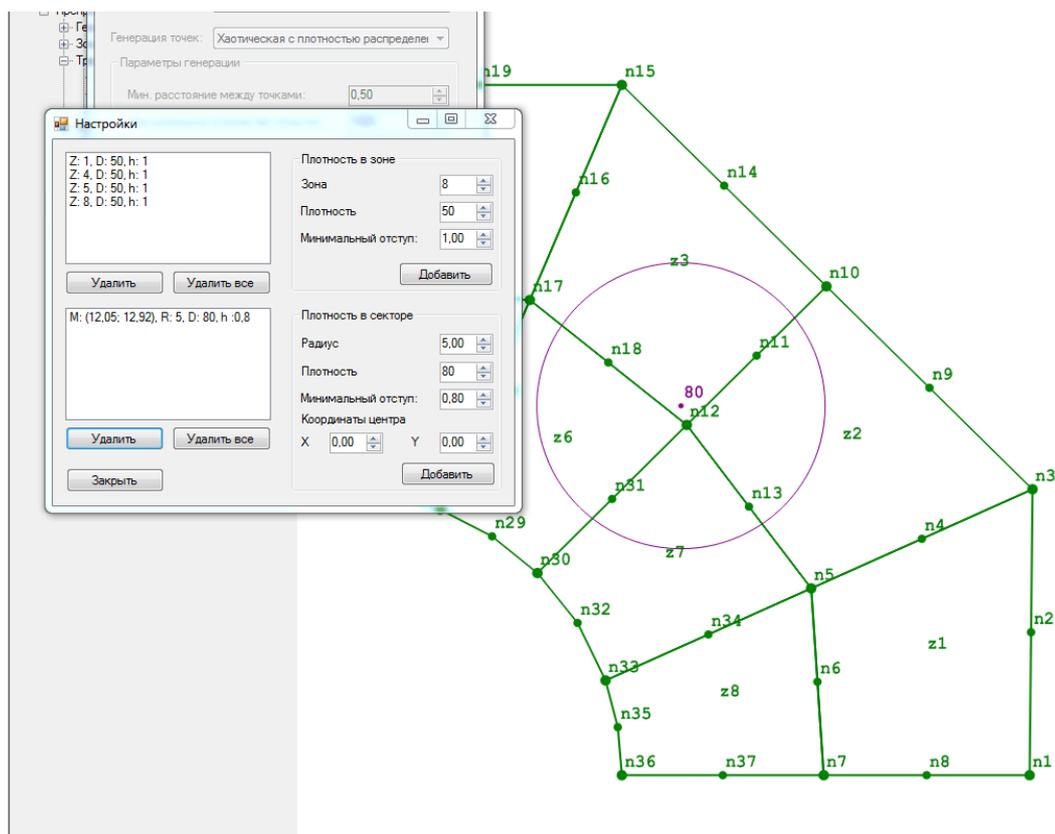
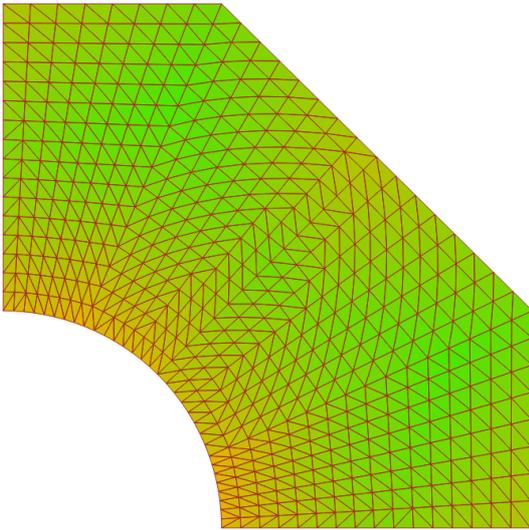


Рис.15. Настройки генерации точек для триангуляции по Делоне.

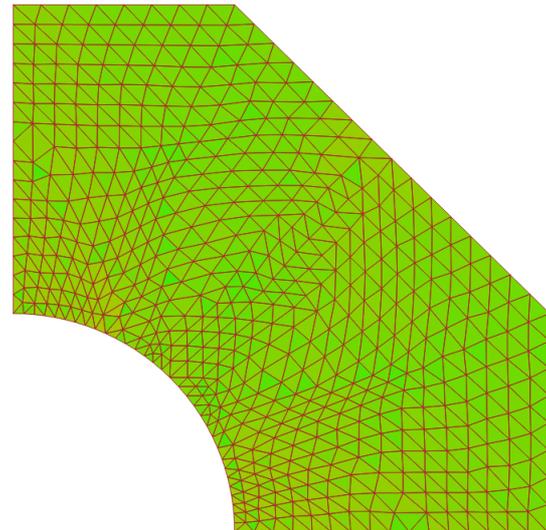
Модуль оптимизации сетки реализует два метода оптимизации: метод увеличения минимального угла в «звезде» сетки и метод регуляризации.

Звездой узла будем называть совокупность конечных элементов, одним из узлов которых является данный узел. Алгоритм известный и опубликован уже давно. На рис. 16. Представлен пример работы алгоритма. Слева сетка до оптимизации. Минимальный угол в КЭ в ней равен $24,69^\circ$. Справа – после оптимизации. Минимальный угол стал равен $36,95^\circ$.

Идея метода регуляризации (рис. 17) заключается в позиционировании каждого узла по центру тяжести многоугольника, образованного его звездой. Алгоритм реализует изменение координат узла относительно многоугольника, который образуют элементы, в составе которых встречается искомый узел.



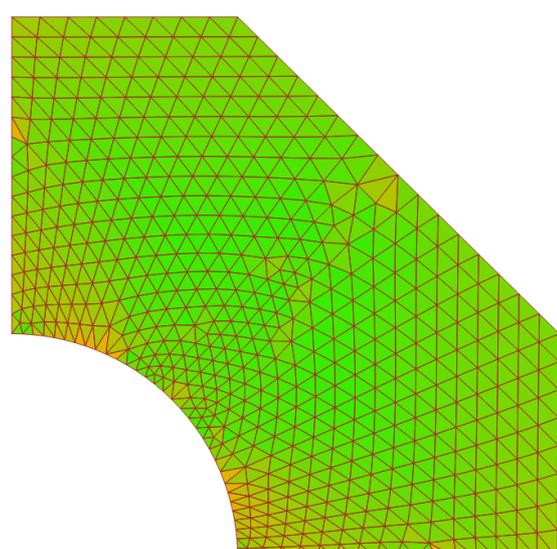
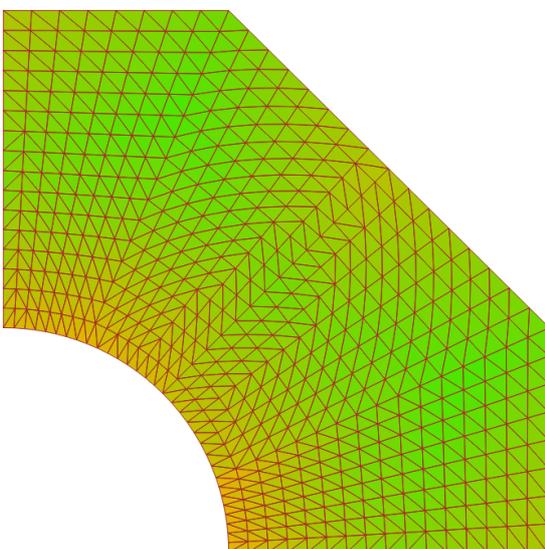
До оптимизации.



После оптимизации.

Рис. 16. Демонстрация работы алгоритма оптимизации сетки.

Координаты находятся по формуле центра масс выпуклого многоугольника. Формула достаточно громоздка и для получения результата необходимы существенные затраты машинного времени. Алгоритм выполняется в цикле для всех узлов до тех пор, пока минимальное перемещение узла не станет меньше заданного достаточно малого ϵ или пока количество итераций не достигнет 1000.



До регуляризации.

После регуляризации.

Рис. 17. Демонстрация работы алгоритма регуляризации сетки.

В подсистеме задания граничных условий реализованы простые закрепления от перемещений выбранных узлов в двух или каком-либо одном направлении. Закрепления задаются на основе линий. Для задания закреплений необходимо указать линию, и система автоматически определит узлы КЭ, которые лежат на этой линии (рис. 18). Система также предоставляет возможность выбрать закрепляемые линии кликом по рабочей области.

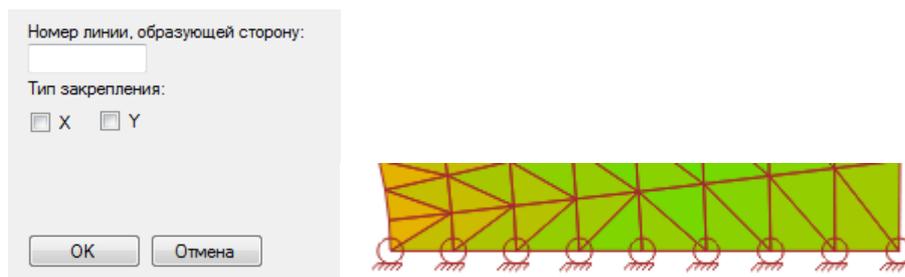


Рис.18. Подсистема реализации граничных условий. Справа – визуализация.

В подсистеме задания внешних воздействий можно реализовать как отдельные силы, так и распределенные по любому закону внешние нагрузки. Согласно концепции МКЭ распределенные нагрузки приводятся к узловым силам. При этом необходимо, чтобы силы, приложенные в узлах, находящихся в зоне приложения распределённой нагрузки, создавали в целом такое же эквивалентное воздействие, что и теоретически распределённая нагрузка.

В препроцессоре распределение сил по узлам проводится исходя из следующего принципа: узловые силы эквивалентны внешним распределённым нагрузкам, если имеет место равенство их работ на возможных перемещениях.

Простейшее соблюдение этого принципа приводит к следующей схеме – к узлу прикладывается сила, равная интегралу от функции нагрузки в пределах от середины расстояний от данного узла до двух соседних нагруженных узлов, если все три узла лежат на одной прямой. В том случае, если данный узел имеет только один соседний нагруженный (нагрузка распространяется только до данного узла), то сила, приходящаяся на данный узел равна интегралу от функции нагрузки в пределах от данного узла до середины расстояния до другого соседнего нагруженного узла.

Графически это можно представить (рис. 19) заштрихованными площадями S_1 для узла, имеющего два соседних нагруженных узла, и S_2 для узла, имеющего только один соседний нагруженный узел.

Рис. 20. Модуль реализации нагрузок. Справа – визуализация.

Для анализа строк выражений, составленных из стандартных математических операторов $+$, $-$, $/$, $*$, функций \sin , \cos , скобок, применяется библиотека, с помощью которой можно переводить выражения из строковой формы инфиксной записи в обработанное представление, составленное в постфиксной нотации, и затем вычислять их.

В препроцессоре реализована библиотека материалов с возможностью добавления новых материалов, их редактирования, а так же удаления. Это поддерживает работу модуля назначения материалов (рис. 21).

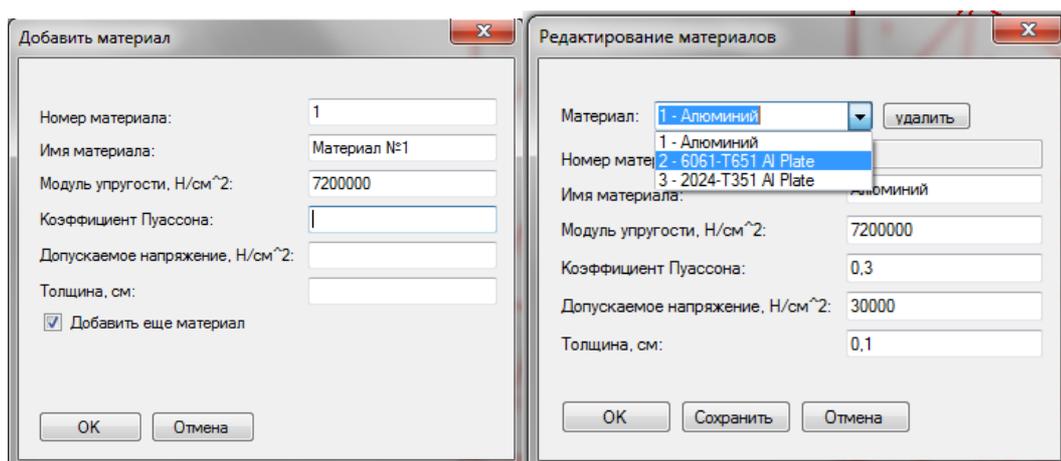


Рис. 21. Модуль назначения материалов. Слева – добавление нового материала, справа – редактирование имеющегося в базе.

Материалы назначать можно как группам КЭ, объединенных зонами, так и отдельным КЭ. При этом материалы, используемые в модели, отображаются на экране вместе со всеми свойствами. Любой из используемых материалов можно легко удалить из модели. При этом все КЭ с этим материалом будут помечены как КЭ с заданными свойствами (рис. 22).

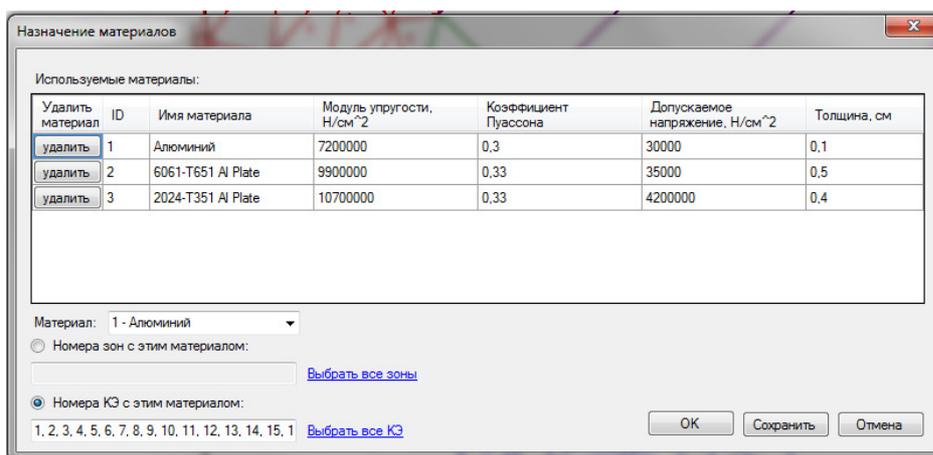


Рис. 22. Назначение материалов КЭ:

Модуль анализа качества сетки КЭ

Для препроцессора разработан мощный модуль анализа сетки конечных элементов. Основные возможности модуля:

- поиск КЭ, содержащих слишком малые или слишком большие углы (пороговые значения минимально и максимально допустимых углов задаются пользователем);
- поиск КЭ с площадью превышающей максимально допустимую. Этот порог также задается пользователем;
- визуальное отображение результатов анализа: выделение цветом «плохих» и «хороших» КЭ;
- обзор полной статистики по сетке;
- отображение информации по конкретному КЭ при наведении на него мышкой в рабочей области (рис. 19):

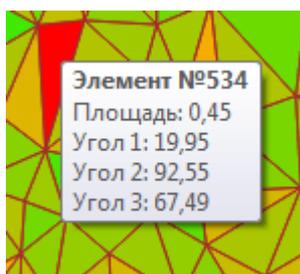


Рис. 23. Информация об отдельном конечном элементе

Пример полной статистики по сетке выглядит так:

«Статистика по сетке: Сетка №3.

Заданные параметры: *Минимально допустимый угол: 20*

Максимально допустимый угол: 120

Минимально допустимая площадь: 0,2

Результаты анализа : *Всего конечных элементов: 774*

Из них содержат недопустимых углов: 35 (2,4% от общего числа).

Элементов с углом меньше допустимого: 3 (0,21% от общего числа).

Элементов с углом больше допустимого: 32 (2,19% от общего числа).

Минимальный угол: 1,6, в элементе: 1101.

Максимальный угол: 176,58, в элементе: 912.

Элементов с площадью меньше допустимой: 451 (30,93% от общего числа).

Минимальная площадь: 0,01, у элемента: 912

Максимальная площадь: 1,19, у элемента: 635».

Модуль анализа является незаменимым помощником в ходе выполнения исследования по определению качества сетки, позволяет оперативно отслеживать сбои в работе алгоритмов, сравнивать различные методы оптимизации, находить ошибки в построении сетки КЭ.

Управляющий элемент модуля анализа показан на рис. 24.

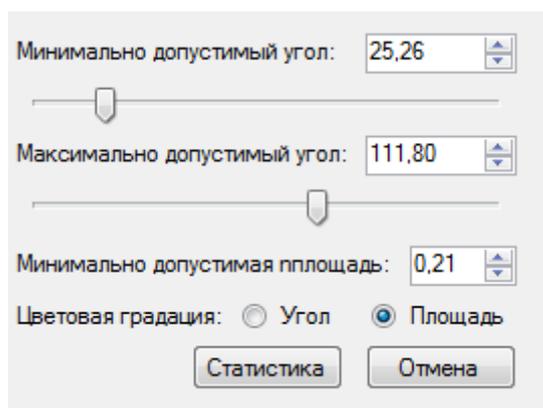


Рис. 24. Управляющий элемент модуля анализа сетки

Выводы:

Разработан мощный инструмент триангуляции плоской области, оптимизации и регуляризации сеток, анализа результатов триангуляции. Возможности препроцессора позволяют его использовать не только в учебных целях, но и как и инструмент в научных исследованиях. В частности, опытная эксплуатация позволила прийти к следующему предварительному выводу: результат триангуляции в большей степени зависит от выбранного метода генерации точек, нежели от выбранного алгоритма триангуляции. Исследования продолжаются.

В процессе эксплуатации препроцессора совместно с системой SIGMA накоплен большой методический материал, позволяющий проводить обучение не только формально-логическими умениям и навыкам, но и обеспечивать развитие творческих и профессиональных способностей разработчиков САЕ-систем, а также ускоренное накопление инженерного опыта с учетом специфических требований проектных организаций аэрокосмической промышленности.

Библиографический список

- 1.Столярчук В.А. Программная система для обучения разработчиков САЕ-приложений. – Научно-технический вестник Поволжья, №6, 2012, стр. 394-397.
2. Бродский А.В., Столярчук В.А., Ульянов С.А. Разработка обучающих подсистем САПР, Учебное пособие, МАИ, Москва, 1990, 68стр.
- 3.Панфилов А.А. Preprocessor: система подготовки исходных данных для САЕ Sigma., Сборник тезисов докладов молодежной научно-практической конференции «Инновации в авиации и космонавтике --2012», Москва, 2012, стр.207-208.
- 4.Ruppert J. A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation, NASA Ames Research Center, Submission to Journal of Algorithms, №4, 1994, стр. 124-138.
- 5.Скворцов А.В. Триангуляция Делоне и её применение, Издательство Томского Университета, 2002, 56стр.
- 6.Sinclair D. S-hull: a fast sweep-hull for Delaunay triangulation. Invented by Dr. David Sinclair, Copyright © 2010, <http://www.s-hull.org>
7. Bourke P. Triangulate: Efficient Triangulation Algorithm Suitable for Terrain Modelling or An Algorithm for Interpolating Irregularly-Spaced Data with Applications in Terrain Modelling. Presented at Pan Pacific Computer Conference, Beijing, China, January 1989, <http://paulbourke.net/papers/triangulate/>